Subject: Re: Keyword precedence
Posted by Martin Schultz on Tue, 29 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

```
Craig Markwardt wrote:
>
> "Mark Hadfield" <m.hadfield@niwa.cri.nz> writes:
>
>> It's interesting that David Fanning and Martin Shultz have both recommended
>> the following idiom for establishing overridable defaults
>>
>> pro my_plot, COLOR=color, _EXTRA=extra
>>     if n_elements(color) eq 0 then color = 12
>>     plot, COLOR=12, _EXTRA=extra
> ****      Ooops  ^^     ****
>> end
>>
>> This has the effect, unintended and normally irrelevant, that if the
>> following call is made with the COLOR keyword set to an undefined variable
>>
>> my_plot, COLOR=color
>>
>> then this variable is set to 12 on output. It isn't too hard to imagine a
>> situation (successive calls to different routines with different default
>> colours) where this will bite an unwary programmer, though in several years
>> of using this idiom I have seldom thought about this side-effect and have
>> very seldom been bitten.
>
> I have had a difficult time keeping up with this thread.  Whew!  I
> often do my keyword passing with the following draconian but safe
> technique.
>
> pro my_plot, COLOR=color0, _EXTRA=extra
>    if n_elements(color0) eq 0 then color = 12 $
>    else color = floor(color0(0))
>    plot, COLOR=color, _EXTRA=extra
> end
>
> COLOR0 is the value passed in, which is distinct from the value of the
> local variable COLOR.  I agree. It's ugly.
>
> Craig
>
> --
>  ---------------------------------------------------------- --------------
> Craig B. Markwardt, Ph.D.        EMAIL:    craigmnet@cow.physics.wisc.edu
> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
>  ---------------------------------------------------------- --------------
```

No, it's not ugly, it's utmost correct ;-) This is what I do
whenever I get caught by the situation that Mark points out -
once I discover that my return value is changed *and* that this
leads to undesired consequences (which most often it does not,
rather the opposite), then I change color to color0 or whatever.
To give you an example, where I rely on setting the keyword value
if it is undefined:

```
pro whatever, filename

    read_data, filename, data
    if n_elements(data) eq 0 then return
    print, ' Read data from file '+filename
    plot,data.x,data.y
end
```

Here, read_data will receive an undefined value if you pass no
filename to whatever. It then sets filename to a default search
pattern (e.g. '*.nc') and calls the dialog_pickfile to select a
file. The name of the file that is selected is stored in filename
for future reference (in this example, the print statement).
Alternatively, if you pass a fully qualified filename, the file
is opened with no further questions, or if you know a little more
about the file, you can pass a search mask like
'/home/myself/data/global*.nc' that will be used as filter for
dialog_pickfile. I find that this works very nicely and veeeery
conveniently in 99% of all cases - just occasionally if I want to
loop over several files at once and have them all "hand-picked",
then I will have to re-initialize filename each time.

Cheers,
Martin

--
 [[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie
[[
[[Bundesstr. 55, 20146 Hamburg
[[
[[phone: +49 40 41173-308
[[
[[fax: +49 40 41173-298
[[
[[martin.schultz@dkrz.de
[[

[illegible]